ROBOT ARENA

GAME AI – Spring 2011
Luis – Dani – Peng
The Idea

• Two robots face off in a futuristic Wild West
• Two player game, each player in charge of designing their own robot
• Design consists of buying equipment and AI components
• Design as Gameplay
• Inspired by the “real life” Robot Wars
The Engine

- Game is based in Unity Engine
- Scripting is in C#
- Pros:
  - Graphics Handling
  - Physics Engine
  - API available
- Cons:
  - Learning curve
  - 3D Complexity
  - Game State complexity
The Agent

• A 3D Robot with IK Bones
• The agent has 7 animations to control
• In Unity its collider is a Capsule
The Agent – High Level States

• 3 High level states:
  – IDLE
  – MOVING (walk, run, evading)
  – ATTACK (depends on weapons)

• Each state must handle model animation and transitions
Game Assets

- Arena
- Robots
- Walls
- Artificial Intelligence
- Weapons
  - Blaster Bullets
  - Grenades
  - Power Of Fist
  - Shield
- Repair Stations
- 100 $ start credits
Weapons

• Why?
  ○ Make game more interesting and cool

• Why so many???
  ○ Make AI decisions interesting and complex
  ○ Different weapons required in different game situations
Blaster Bullet

- Graphics: Sphere with Smoky Trail
- Damage: Reduces HP by -5
- Cost: 20 $

- Infinitely Long Range weapon
- When enemy is sighted far away
- Does least damage
Grenades

- Graphics: Actual Grenade model
- Damage: Reduces HP by -20
- Cost: 55 $

• Medium Range Weapon

• When attack over obstacles

• In situations where enemy is hiding behind walls
Power Of Fist

- Graphics: Ball of Fire from Hands
- Damage: Reduces HP by 40
- Cost: 75 $

• Very short range weapon
• In situations when really close to enemy
• Hence does most damage
**Shield**

- Graphics: A transparent sphere
- Heal: Shields robot from weapons
- Cost: 30 $

- Only lasts for 5 seconds per activation
- A weapon from p.o.v of protection
- Limitation on time for fairness
Repair Station

- Graphics: Health Cross
- Heal: Increases HP by 50

- Recover from damage
- Makes game interesting
- Helps in making certain AI decisions
AI Design

Two Principles:

1. Equipment ↔ AI

2. Keep It Simple, Understandable
AI Components

• Original design: Algorithm-centric
  – Players choose between algorithms like A* and DFS for path finding, Minimax and Decision Trees for decision making, etc.

• Problem: Algorithmic choices were not obvious to the player

• Current design: Behavior-centric
  – Players choose between explicit behaviors and conditions in an Action-Condition Graph
AC Graph Example

MoveToEnemy

EnemyInRange: Grenade
- EnemyInRange: Blaster
  - ThrowGrenade
  - FireBlaster
AC Graph Execution

- Directed Graph Traversal
- One Action Node executed per Update()
- Multiple Condition Nodes Evaluated per Update()
AC Graph Creation

• Currently, a text-based interface:
  1. MoveToEnemy(2)
  2. EnemyInRange: Grenade(T:3, F:4)
  3. ThrowGrenade(2)
  ...

• Ideally, a “circuits” interface with nodes and connections
Conclusion and the Future

- Equipment ↔ AI an interesting idea
  - Add the algorithmic aspects back in at some point
- Many possible game scenarios
  - Tournament
  - KOTH
  - Random Draft
- Making the design process easy for players a major priority
Trade-offs

• User Interface:
  – Our original design had a user interface where user can buy different weapons, assets for the Robot
  – Also a user interface for AI graph-node design

• Problems:
  – UI would be too complex to implement in Unity; therefore, difficult to debug